REMARKS/ARGUMENTS

This letter is in reply to the office action dated November 21, 2008.

As a preliminary note, the Applicants appreciate that the Examiner is not bound by decisions of patent offices in foreign jurisdictions. However, should it be of interest to the Examiner, Applicants' corresponding application in Europe has now issued to patent as European Patent No. 1,570,347 B1. The references currently relied upon by the Examiner were also duly considered by the European Patent Office.

To expedite prosecution of the application, a number of amendments have been made to the claims as will be discussed below. Applicants have cancelled claims 6, 12 and 18 without prejudice. Accordingly, claims **1-5, 7-11, 13-17 and 19-21** remain pending in this application. Claims 1, 7, and 13 are independent.

Amendments to the Claims

In the claims, Applicants have replaced the term "cod file" with the term "generated file". These amendments are supported by paragraph 16 in the application as filed.

Applicants have amended claim 1 to specify that one or more files can be generated from a plurality of class files. This amendment is supported by, for example, paragraphs 23, 29, 31 and 42 in the application as filed. Applicants have also deleted the wording "without duplication of entries" when describing how the byte codes and information structure is generated. Further, to better define the structure of the device, claim 1 has been amended to recite "a computing unit connected to the memory unit...the computing unit configured to execute software for generating...".

Applicants have amended claim 2 to better describe the nature of the fixup table. These claims now recite that the information in the fixup table comprises the location of data needed for resolving a symbolic reference in the given generated file. Support for this

claim amendment may be found at, for example, paragraphs 25, 35 and 42 of the application as filed.

Amended claim 3 recites that there are at least two generated files defined as sibling files in a common sibling group, and that each of the sibling files comprise a sibling list for listing other sibling files in the common sibling group. Further, claim 3 now recites that cross-references between the sibling files in the common sibling group are indicated using hard offsets and references to files that are not part of the common sibling group are indicated using symbolic references. Support for this claim amendment may be found at, for example, paragraphs 31, 33, 34 and 41 of the application as filed. New claim 19 recites that the information in the fixup table of a given sibling file comprises the location of data of a cross-referenced sibling file to place one of the hard offsets that corresponds to the cross-referenced sibling file into context at link time. Support for this claim amendment may be found at, for example, items 3 and 4 in paragraph 42 of the application as filed.

Applicants have amended claim 4 to properly refer to the antecedent "given generated file" and have added the word "second" before the term "hard offset" to distinguish from previous use of the term "hard offset" in some of the claims that claim 4 depends on.

Applicants have amended claim 5 to recite that at least one of the hard offsets does not need to be resolved by the Java Virtual Machine at link time. This amendment is supported for example, by paragraph 23, and items 1 and 2 in paragraph 42 of the application as filed.

Amendments have been made to claims 7-11 that are analogous to amendments made to claims 1-5, respectively. Amendments have also been made to claims 13-17 that are analogous to the amendments made in 1-5, respectively. New claims 20 and 21 have been added, reciting analogous features claimed in claim 19.

Further, claim 13 has been amended to recite "[a] computer-readable medium storing computer executable instructions" and claims 14-18 and 21 which depend on claim 13 have been amended to recite "[a] computer-readable medium".

Accordingly, claims **1-5, 7-11, 13-17 and 19-21** are currently pending in this application. Claims 1, 7 and 13 are independent.

Claim Objections

Previous claims 2, 8, and 14 were objected to, as the limitation "the cod file" had no explicit antecedent basis. Claims 2, 8, and 14 have been amended to recite the limitation "the given generated file" accordingly.

Previous claim 7 was objected to, as the limitations "constant pool entries from the class files" and "information structure entries from the class files" had no explicit antecedent basis. Claim 7 has been amended to recite "constant pool entries from the plurality of class files" and "information structure entries from the plurality of class files" accordingly.

In view of the amendments made to claims 2, 7, 8, and 14, withdrawal of the objections to these claims is respectfully requested.

Claim Rejections - 35 U.S.C. §112

Previous claims 1 and 7-12 stand rejected under 35 U.S.C. §112 as being indefinite.

With respect to previous claim 1, the Examiner opines that it appears to be reciting intended use in a method step within a device claim. Claim 1 has been amended to recite "a computing unit connected to the memory unit...the computing unit configured to execute software for generating...". In view of this amendment, the Applicants respectfully submit that the structure of the device is now clear.

With respect to previous claim 7, the Examiner notes that the limitation "in at least one of the constant pool and the byte codes and information structure" has insufficient antecedent basis. Claim 7 has been amended to recite "a plurality of class files having constant pools" and "in the constant pools of the class files". In view of this amendment, the Applicants respectfully submit that the limitation has proper antecedent basis, and that claims 8-12 which depend on claim 7 are now clear.

In view of the amendments made to claims 1 and 7 noted above, withdrawal of the rejections under 35 U.S.C. §112 is respectfully requested.

Claim Rejections - 35 U.S.C. §101

Previous claims 7-18 stand rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter.

Claim 7 has been amended to recite "A method for generating one or more files from a plurality of class files having constant pools on a computing unit...". In view of this amendment, the Applicants respectfully submit that claims 7-12 are directed to statutory subject matter and that the Examiner's concerns have been addressed.

Claim 13 has been amended to recite "[a] computer-readable medium storing computer executable instructions". Claims 14-18 and 21 which depend on claim 13 have been amended to recite "[a] computer-readable medium". In view of these amendments, the Applicants respectfully submit that claims 13-18 are directed to statutory subject matter and that the Examiner's concerns have been addressed.

In view of the amendments made to claim 7 and claims 13-18 noted above, withdrawal of the rejections under 35 U.S.C. §101 is respectfully requested.

Claim Rejections - 35 U.S.C. §103

Claims 1-18 stand rejected under 35 U.S.C. §103(a) as being unpatentable over WO 99/49392 in the name of Baentsch et al. (hereinafter referred to as Baentsch) in view of U.S. Publication No. 2002/0170047 A1 in the name of Swetland. The Applicants respectfully traverse all rejections.

There is no clear motivation for someone skilled in the art to combine Baentsch and Swetland

The Applicants respectfully submit that there is no clear motivation for someone skilled in the art to combine Baentsch and Swetland.

Baentsch appears to teach a CAP file format comprising a text section, a data section, and a fixup table, also known as a modified constant pool, for indicating relocation information. The CAP file format divides the modified constant pool into two parts: internal information for use only during linking and external information to be preserved for late code binding. The internal information is removed from the modified constant pool after linking.

In particular, it appears that Baentsch is only directed towards the resolution of references (see line 21 on page 1 of Baentsch). It does not appear that Baentsch is concerned with combining several classes into one single object. Rather, Baentsch is concerned with improving the linking process for resolving references into well-known and trusted packages (see lines 13-14 on page 4 of Baentsch), and accordingly deals with IBM or JavaSoft software library files (see lines 24-31 on page 3 of Baentsch). These are standard library files, which one typically does not merge files with. Further, Baentsch does not teach using the CAP file formats with devices having memories with limited storage capacity. Accordingly, there is no need in Baentsch to even consider combining several classes into one object.

Swetland appears to teach a unified programming object with a shared constant pool comprising global constant pool entries mapped from local constant pool entries for two or more class files; and, a plurality of object code copied from the two or more class files to the unified programming object and identified by the constant pool entries.

In addition, although Swetland teaches creating a single unified programming object by combining two or more class files without duplication of entries in the constant pool, it does not appear that Swetland recognizes the problem that in some cases devices have memory units with a limited storage capacity, and therefore may have limits on the size of individual files. Accordingly, it may not be possible to combine all of the class files into a single file. In contrast, the Applicants clearly recognized this situation in paragraphs 29 to 31 of the application as filed.

Furthermore, in Swetland, the entries in the constant pool are of fixed-length and are assigned slot numbers. A relocation that needs to be resolved is located by referencing a constant pool slot and reading the offset provided by the entry in the constant pool slot (see paragraph 0085, left column, lines 3, and 16-18 in Swetland). Accordingly, it appears that the offset disclosed in Swetland is not directly addressable.

In addition, Swetland does not provide any teaching regarding the use of a fixup table. There is no mention of a fixup table in reference Swetland because the 'bundle' format, as described, is an extension of the existing class file format and therefore contains no explicit fixup information. All locations in the executable byte codes and information structure that could require fixups are encoded in terms of references to the constant pool. Part of a traditional Java Virtual Machine's job of loading class files involves re-writing these references in terms of the results of resolving the constant pool entries.

Based on the above observations, the Applicants respectfully submit that there is no clear motivation for someone skilled in the art to combine Baentsch and Swetland. Baentsch is simply directed towards resolving references during the linking process to well-known and trusted target packages. Swetland is simply concerned with removing

redundant entries in the constant pools of several files by combining the files into a single unified programming object and removing the redundant entry in the constant pool of the single unified programming object.

Accordingly, the Applicants respectfully submit that Baentsch and Swetland are directed towards different applications, and one skilled in the art would not think to combine the two references for the purpose of the Applicants' claimed embodiments, which is to generate one or more files from a plurality of class files in which the generated files have a reduced file size. The solution of providing information for resolving references at link time via fixup tables (Baentsch) is unrelated to the concept of combining of class files (Swetland). It is quite clear that since the techniques taught by Baentsch and Swetland are not related to one another, that one skilled in the art would not think to combine the two references for the purpose of the claimed invention.

Further, the Applicants submit that because references Baentsch and Swetland address different issues which are unrelated to one another, neither of these issues can be inferred from reading the other of the references, and so a person skilled in the art, when reading one of these references, will not have any motivation to search for the other reference.

> *Even if Baentsch and Swetland were combined, a person skilled in the art without inventive ability would not have been able to obtain to arrive at the Applicants' claimed embodiments*

Additionally, the Applicants respectfully submit that even if the teachings of Baentsch and Swetland were combined, a person skilled in the art without inventive ability would not have been able to obtain to arrive at the claimed invention. Specifically, Baentsch and Swetland cannot be combined since they teach techniques which are in direct conflict with one another. Baentsch teaches resolving references during the linking process by using information in a fixup table. Meanwhile, Swetland does not teach the use of a fixup table because the 'bundle' format, as described and used in Swetland, is

an extension of the existing class file format and therefore contains no explicit fixup information. All locations in the executable byte codes and information structure that could require fixups are encoded in terms of references to the constant pool. Part of a traditional Java Virtual Machine's job of loading class files involves re-writing these references in terms of the results of resolving the constant pool entries. Applying fixups is generally a less expensive operation than the traditional technique. Fixups list exactly what needs to be changed. Meanwhile, the traditional method, used by Swetland, requires the Java Virtual Machine to visit every entry in the byte codes and information structure to see if any entries require re-writing in terms of a resolve constant pool entry.

Summary

Amended claim 1 of the present application recites a device with a computing unit that is configured to execute software for generating one or more files from a plurality of class files (by combining elements from the plurality of class files without duplication of entries for reducing storage space). Amended claim 1 further recites that the number of the generated files is less than the number of the plurality of class files, and a given generated file comprises: a constant pool created by combining constant pool entries from two or more of the plurality of class files without duplication of entries; a byte codes and information structure created by combining byte codes and information structure entries from the two or more of the plurality of class files; and a fixup table for providing information to the Java Virtual Machine for resolving at least one entry in the given generated file at link time.

Amended claim 1 recites creating one or more generated files while Swetland teaches creating a single unified programming object by combining two or more class files without duplication of entries in the constant pool. Swetland does not recognize the problem that in some cases, devices have memory units with a limited storage capacity such that combining all of the class files into a single file is not possible.

Accordingly, the Applicants submit that combining Baentsch and Swetland does not achieved the claimed subject invention since the combination of Baentsch and Swetland fails to realize the solution of generating one or more class files to address the technical problem of storage devices with limited storage capacity and therefore individual file size limits. Relying on Swetland, one can clearly exceed a file size limit since Swetland teaches always combining all class files into one file.

In view of the foregoing, the Applicants respectfully submit that amended claim 1 defines subject matter that is both novel and inventive over the cited references, and is now in form for allowance. It is respectfully submitted that amended claims 7 and 13 also define patentable subject matter for at least the same reasons. It is further submitted that the dependent claims also define patentable subject matter for at least the same reasons. Withdrawal of the rejections under 35 U.S.C. §103 is respectfully requested.

In view of the foregoing comments, it is respectfully submitted that **claims 1-5, 7-11, 13-17 and 19-21** are now in condition for allowance, and a notice to that effect is respectfully requested.

Respectfully submitted,

BERESKIN & PARR
Agent for the Applicants

By _____
Kendrick Lo
Reg. No. 54,948
Tel: 416-364-7311